Comparing the Performance of Machine Learning Algorithms for Emotion Classification on Tweets

Sugeeshwa S P Galhena,[1] Ajantha S Atukorale[2]
University of Colombo School of Computing, Colombo, Sri Lanka
[1]sugeeshwag@gmail.com,[2]aja@ucsc.cmb.ac.lk

***Abstract* — The rapid increase in the availability of textual content due to Industry Revolution 4.0 has made sentiment analysis an important area of machine learning research. This study aims to develop a mechanism to identify the hidden emotions in textual content, beyond the three basic sentiments of positive, neutral, and negative. Several machine learning approaches to emotion classification, including Naive Bayes classifiers, Support Vector Machines, Regression, Decision Trees, and Random Forests have been explored. The experiments show that simple linear models can achieve high accuracy (up to 90.5%), suggesting that complex algorithms are not always necessary for effective emotion classification. The performance of the models was evaluated using a variety of metrics, including accuracy, precision, recall, F-score and efficiency. The findings suggest that machine learning approaches can be used to effectively identify emotions in textual content, even with simple models. This has potential applications in a variety of domains, such as social media analysis, customer service, and healthcare.**

Keywords* — *emotion detection, sentiment analysis, machine learning, supervised learning, text classification

I. Introduction

Detecting emotion in the text is a field closely related to sentiment analysis. While sentiment analysis aims to uncover the three opinions the text contains such as positive, negative, or neutral, emotion analysis focuses on discovering the feeling behind the text such as anger, fear, joy, sadness, etc. Reading an online review to decide on a novel to read, or an item to purchase online could be considered as examples of sentiment analysis one faces during day-to-day life. Many studies have been conducted throughout the years to classify these emotions. Humans feel a spectrum of emotions, as opposed to the three basic sentiments positive, negative, and neutral which were discussed above. In real life, sentiment does not have an impact on the evaluation of emotions. For example, book reviews "That book was awesome!" and "That book was good" would be of positive sentiment, but the emotionality behind the two reviews has two different meanings.

[1] and [2] have conducted a thorough research on approaches to emotion detection from text with emotion models, and approaches. The latter categorized approaches based on classification techniques that can be used to identify emotions. The former did a critical analysis of the approaches and suggested that both lexicon-based approaches, as well as machine-learning approaches, can be used in the analyses. By comparing the results with other factors, the best algorithm that provides better results is selected. Indicators such as recall, precision, accuracy, confusion matrix, F1 values, etc. are calculated to measure the performance of the algorithm. [3] proposed to calculate both accurate and unweighted measurements.

This study aims to review and analyze different methodologies and techniques used for text-based emotion detection. The study by [1] consists of an extensive survey to differentiate between the advantages and disadvantages of emotion classification models and as well as the approaches that have been utilized in the past for text mining. They recommended that depending on the aims and the needs of the study, lexicon-based approaches or machine learning-based approaches can be applied in extracting emotion from the text.

To accomplish these objectives, appropriate hypotheses are being proposed based on past research. Furthermore, outcomes of the intermediate studies are utilized to make the inferences. By investigating the existing approaches, this paper aims to provide insights into the current state of the field and identify potential areas for improvement and future research.

Correspondence : Sugeeshwa S P Galhen (e-mail : sugeeshwag @ gmail .com) Received : 08 - 01 - 2024 Revised : 17 - 03 - 2024 Accepted : 20-03-2024

Sugeeshwa S P Galhena and Ajantha S Atukorale are from University of Colombo School of Computing, Sri Lanka (sugeeshwag@gmail.com, aja@ucsc.cmb.ac.lk)

DOI: https://doi.org/10.4038/icter.v17i1.7277

II. Related Work

A. Emotion Classification Models

Over the years, different emotion classification methods have been introduced, categorized into emotion dimensions and emotion categories. Emotion dimensions provide more information about the emotions such as the positivity or negativity of the emotion, arousal of the emotion, and degree of the emotion while emotion categories are quite popular among machine learning researchers due to its simplicity. These concepts are explained extensively in the paper [4] as Dimensional Emotion Models (DiEMs), where emotions are located in a special space that suggests a dependency between the emotions, and Discrete Emotion Models (DEMs), where the emotions are located in different categories.

Multiple DiEMs discussed in the paper [4]. Rusell's Circumplex Model of Affect which suggests that emotions are in the arousal - valence dimensions, and Plutchik's 2D Emotion of Wheel. The latter proposes a model with arousal on the horizontal axis and valence on the vertical axis. Rusell & Mehrabian's Model suggests that the emotions are distributed in a 3D space according to arousal, valence, and dominance. In the study [5], the researchers did a similar study about emotional combinations on content that goes viral. The researchers considered more than 65,000 articles in their study and concluded that it is not the emotion that makes content go viral, but where the emotions fall within the Valence-Arousal-Dominance (VAD) model. As stated in the article, an emotion is a combination of three characteristics : Valence, Arousal, and Dominance.

The Paul Ekman Model which suggests six fundamental emotions (anger, disgust, fear, happiness, sadness, surprise), The Robert Plutchik Model which suggests eight primary emotions that occur in contrasting pairs (anger vs fear, joy vs

sadness, surprise vs anticipation, trust vs disgust), and Orthony, Clore, and Collins (OCC) Model which suggests twenty two emotions as opposed to the basic emotions suggested in both Ekman's and Plutchik's models by adding additional classes such as appreciation, envy, pity, relief, shame, etc. are the DEMs that are discussed in the paper [4]. The paper [6] recommends multiple DEMs that have been defined by the psychologists. Two other such models are Shaver Model and Tomkins Model. Shaver Model consists of six emotions: sadness, joy, anger, fear, love, surprise which can be represented in a tree structure while the Tomkins Model comprises of nine emotional states. Out of these nine, only three emotions are positive and most these can be represented as pairs. The emotions he introduced are disgust, surprise-startle, anger – rage, anxiety, fear - terror, contempt, joy, shame, interest – excitement. The paper [7] suggests that Parrot's Model, a DEM, which contains a balanced set of classes: love, anger, sadness, joy, fear and surprise improves the accuracy as compared to the other techniques when it is used on analyzing blog data. Despite the fact that emotional categories may not encompass all emotions, most computer-based methods rely on them since they are reliable and simple to use.

B. Emotion Detection

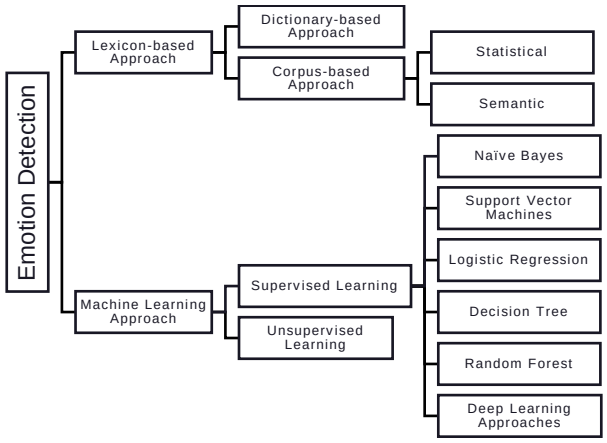

Fig. 1. Computational approaches for emotion detection

The survey, [1] stated that the methods that are used to detect emotions can also be divided into two; lexicon based approach and machine learning based approach. Keyword-based approaches and ontology-based approaches fall into lexicon based approaches. The statistical approach such as Latent Semantic Analysis is also considered as a lexicon based approach. The paper [6], presented an analysis on usage of different machine learning techniques that have been conducted in the past for classifying emotions in text. The Fig. 1 illustrates approaches to solve the problem of emotion detection. Lexicon-based approaches use word dictionaries where each one is assigned with a sentiment value. The accuracy of dictionary-based approaches depends on the algorithm that is used for the analysis. The corpus-based approaches tend to be more efficient in a particular domain because it maintains domain-specific terms.

The machine leaning approaches that are being discussed on the paper are Naïve Bayes, Support Vector Machine (SVM), Decision Trees, etc. These machine learning approaches have been experimented on the studies such as [8], which applied the SVM on a collection of text collected via customer feedback surveys, [9] which applied Naïve Bayes, SVM, and n-gram model in classifying online reviews about seven popular travel destinations around the world. The second experiment claimed to achieve more than 80% calculation accuracy for all the three models. Other methodologies that are being discussed in the paper [6] are deep learning approaches and hybrid approaches in classifying textual content. It is stated that the hybrid approaches such as Random Forest Support Vector Mechanism (RFSVM) has been able to perform well. [10] experimented to recognize fifteen different emotions which could be indicative of suicidal behavior using binary SVM classifiers. In their study on emotion detection using deep learning approaches, [3] discussed a few challenges one would face in the NLP research area are word-sense disambiguation (E.g. "Shut up!") and coreference resolution. Thus, the article suggests doing experiments with different machine learning models to conclude. The study [11] carried out a survey on mining and classifying customer reviews, which is another aspect of text classification, opinion mining. The paper suggested supervised, unsupervised and semi-learning methods on text classification. Out of the research that the paper has surveyed, it is mentioned that the Naïve Bayes is one of the commonly used and has performed efficiently in a study that was carried out for classifying Urdu and English opinions in a blog.

The paper [12] analyzed two different datasets of movie reviews with multiple approaches including a deep learning approach. Different approaches were taken to analyze data such as Bag-of-Words (BOW) followed by a Random Forest Classifier, BOW followed by SVM, Word2Vec followed by Random Forest Classifier and Word2Vec, Clustering followed by Random Forest Classifier. By applying Recursive Neural Network (RNN), they achieved higher accuracy in predicting the text category. In their study, [13] used a Convolutional Neural Network (CNN) to classify Chinese microblogs. They used a dataset of 1.6 million tweets and conducted several deviations of the CNN models. After multiple experiments, the highest accuracy the model was able to achieve was 86.83%. In the study [14], the researchers introduced various methodologies that one can take to identify the sentiment behind text and its applications. Some of the methods that were discussed in the study are, RNN, CNN, and Long Short-Term Memory (LSTM). The paper [15] proposed a transformation method to convert a multi-level classification problem into a binary classification problem and was solved by suggesting a deep learning approach. The proposed model has been able to outperform the achieving a higher score on the SemEval2018 Task 1: E-c multi-label emotion classification problem.

Another study [16] focused on emotion analysis measured not only the emotion but also the intensity of the emotion in tweets. They claimed it is important for applications to know the degree to which an emotion is expressed in text. Their findings are presented in an interactive visualization [17] which provides the ability for the readers to cross-filter as well. Another research [18] published a paper based on their findings for the "SemEval-2019 Task 3" which was focused on contextual emotion detection in text. Furthermore, in a separate study [19], the researchers proposed a way to improve three-way decisions using fuzzy logic and deep learning. The reason behind this model was to provide a framework when uncertainty is present in the reviews. In their study, they used feature selection methods such as Term Frequency-Inverse Document Frequency (TF-IDF), Best Match 25 (bm25), Uniformity (Uni), and Inverted Conformity

Frequency (ICF). Eight state-of-the-art models were selected and evaluated in the study based on their performance. Model performances were assessed using Precision, Recall, and F-measure. But the results evidenced that the proposed model's performance was better compared to the other models in the two datasets that were considered in the study.

III. Methodology

A. Dataset

The emotion annotated dataset used in [20] is used for this study. This dataset is publicly available for research purposes. The data set is annotated based on the Parrot's Model and the metrics of the dataset are mentioned in TABLE I and the distribution of sample length (i.e., the length of an input record) is depicted in Fig. 2. The word count of the input record is represented by the x axis while the sample count is represented by the y axis. This data set is already broken down into training, validation and test sets.

TABLE I. Emotion Dataset Metrics

Metric Name	Value	
Number of Samples	20000	
Number of Classes	6	
Number of Samples per Class	joy	6761
	sadness	5797
	anger	2709
	fear	2373
	love	1641
	surprise	719

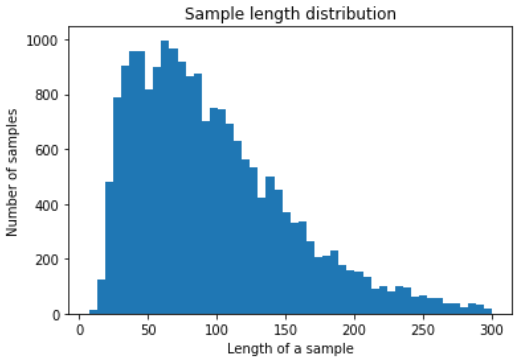

Fig. 2. Sample length distribution.

B. Breakdown of Tasks

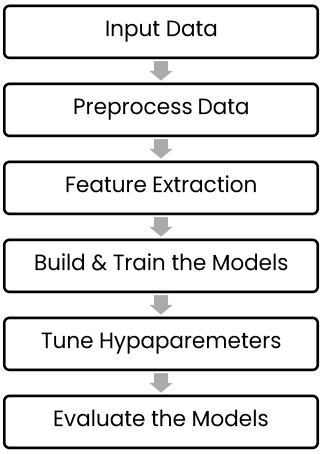

Fig. 3. High-level overview of the workflow

Multiple steps were taken to make sure that the project was a success. Some of the steps are preprocess data, extract features, build and train the machine learning models, tune hyperparameters, and evaluate model performance. Fig. 3 gives a high-level overview of the task breakdown. The model training is broken down into multiple subtasks and various approaches are carried out during these subtasks. Preprocessing is done for the data to make sure that data are in a format that is understandable by the machine learning algorithms. Preprocessing techniques that are carried out are converting all the text into lowercase, removing stop words from the data set, tokenization and vectorization. Several subtasks are carried out during each step such as for vectorization, both BOW and TFIDF methods are carried out. BOW defines a fixed-length count vector, with each entry corresponding to a word in the predefined word dictionary. A word in a sentence is assigned a count of 0 if it is not in the defined dictionary, otherwise it is assigned a count of 1 or higher depending on how many times it appears in the sentence. Although this is an easy to implement method, the drawbacks of it such as does not capture the order of the words, and not considering the meaning of the sentence are impactful when it comes to practical applications [6]. TFIDF represents the text in matrix form, with each number quantifying the amount of information those terms contain in a particular document. The assumption behind this method is that rare words contain more information than the others.

The reason for carrying more than one approach for the subtasks is to compare the results from each task and select the best approach for each task which provides the output with the highest accuracy in less time.

The experiments were carried out in Google Colaboratory in Python environment. Python version 3.7 was used in the study. The relevant libraries were installed and called during the deployment such as: NumPy, Pandas, Matplotlib, Scikit-learn, and Seaborn.

C. Classification Models

1) Naïve Bayes Classifier

This is the most commonly used classifier, and its simplicity has been the reason behind universal usage. Naïve Bayes Classifier computes the posterior probability of a class, based on the distribution of the words in the document.

$$P(label|features) = \frac{P(features|label)*P(label)}{P(features)} \quad (1)$$

2) Support Vector Mechanism

Support Vector Classifier determines linear separators in the search space which can best separate different classes. This is widely used in classification, regression and outlier detection. This was implemented in Python with the help of SGDClassifier.

3) Logistic Regression

Logistic regression is the process of modelling the linear relationship between the dependent variable and the independent variables when the outcome is binary. For this task, the logistic regression model is generalized to multiple classes.

4) Decision Tree

Decision tree is a tree-like structure where a branch represents a decision rule, a node represents an attribute. The nodes at the bottom of the tree are known as leaves and the

topmost node is known as root node. Attribute selection can be done using multiple methods such as information gain, maximum entropy, etc.

5) Random Forest

Random Forest classifier is suitable when dealing with high dimensional noisy data. This is basically a collection of decision trees.

D. Deep Learning Models

1) Convolutional Neural Networks (CNN)

Convolutional Neural Networks is a neural network type which follows a feedforward architecture. Therefore, the connection between two nodes does not perform a cycle. CNNs can perform feature extraction with minimal preprocessing, which is considered as one of the key advantages of the same.

2) Recursive Neural Networks (RNN)

Recursive Neural Networks is a neural network type which is also derived from feedforward architecture. But this differs from CNN as connections between nodes can create a cycle. In this paper LSTM is applied, which is a type of RNN that is effective in modeling sequential data.

The architecture of the deep learning model implementation goes as follows. The implementation is done using Keras, a TensorFlow API.

- Embedding layer: The input text sequences are first passed through the embedding layer. This layer learns and maps each word in the input sequence to a dense vector representation. The embedding layer helps capture the semantic meaning of words and their contextual relationships.
- LSTM layer: The output from the embedding layer is fed into the LSTM layer. This captures long-term dependencies in the input sequences. The LSTM layer consists of LSTM units that process the input sequence and propagate information through time steps.
- Dense layer: The output from the LSTM layer is then fed into the dense layer. This performs a linear transformation on the input and applies an activation function.

In this paper, the activation function used in the dense layer is softmax, which outputs a probability distribution over the possible classes. The model is trained using the categorical cross-entropy loss function and optimized using the Adam optimizer.

E. Hyperparameter Tuning

Hyperparameter tuning is the process of identifying the combination of values which either increases the accuracy of the model or reduces the training time of the model. Selecting the best set of values is an iterative process that depends on time and money. Fig. 4 illustrates the hyperparameter tuning process. Hyperparameter tuning is done not only for the classifiers, but also for the feature selection methods. But hyperparameter tuning is an expensive and lengthy process. Therefore, this was done only for the most vital hyperparameters, and the tuning was done with the objective of increasing accuracy. With multiple trial-and-error experiments, the optimal values for the hyperparameters are identified. Hyperparameters can be tuned using multiple ways: Grid Search and Random Grid Search are two such examples. A search space is defined as a grid for hyperparameters and every position in the grid is searched and evaluated in the grid search. In randomized grid search, a search space is defined as a bounded domain of hyperparameters, and values are evaluated randomly. In this paper, hyperparameter tuning is implemented using grid search.

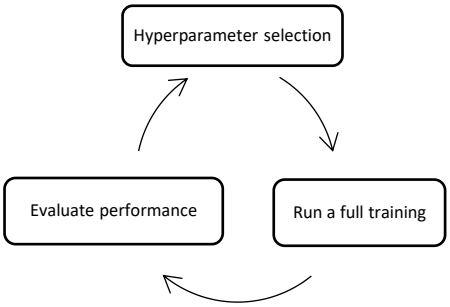

Fig. 4. Hyperparameter tuning process

F. Evaluation

The performance of each model was tested using precision, recall, and F1-score to decide how well the model fitted on unseen data. The terms that are used in the aforementioned matrices are explained below. These performance indicators measure the predictive accuracy of the model.

- TP (True Positives): The number of records correctly classified as belonging to a particular emotion category.
- FP (False Positives): The number of records incorrectly classified as belonging to a particular emotion category.
- TN (True Negatives): The number of records correctly classified as not belonging to a particular emotion category.
- FN (False Negatives): The number of records incorrectly classified as not belonging to a particular emotion category.

1) Confusion Matrix

This is a two-way table which provides the counts of both correct and incorrect predications, based on actual knowns. The confusion matrix also contains TP, FP, TN, and FN.

2) Accuracy

Accuracy measures how often the classifier predicts correctly.

3) Precision

Precision is the accuracy of positive predictions.

4) Recall

Recall is the fraction of the positives that are correctly identified.

5) F1 – Score

F1-score measures what percent of positive predictions are correct.

6) Efficiency

Efficiency measures the time to construct the model, and the time it takes to use the model. This plays a huge role as it is highly costly to run a model for a long time.

IV. Results

After the model has been fitted to the training set and evaluated on the validation set, hyperparameter tuning is carried out on the validation set. This was a repetitive process and the model that performs the best on the validation set is selected and the results were confirmed on the test set. The performance of each model is evaluated based on predetermined evaluation criteria.

TABLE II. Results Before Hyperparameter Tuning

Model	Feature extraction	Accuracy	Weighted average for F1-score
Naïve Bayes Classifier	BOW	78.80%	0.76
	TFIDF	67.75%	0.61
Support Vector Mechanism	BOW	89.85%	0.90
	TFIDF	90.15%	0.90
Generalized Logistic Regression	BOW	89.85%	0.90
	TFIDF	87.30%	0.90
Decision Tree	BOW	88.50%	0.87
	TFIDF	86.70%	0.87
Random Forest	BOW	88.30%	0.88
	TFIDF	88.95%	0.89

Before hyperparameter tuning, the SVM using BOW and TFIDF features performed the best in terms of accuracy and weighted F1-score. The Decision Tree and Random Forest models showed relatively lower performance compared to the SVM and Logistic Regression models, but still achieved respectable accuracy and weighted F1-scores.

TABLE III. Results After Hyperparameter Tuning

Model	Feature extraction	Accuracy	Weighted average for F1-score	Time taken to train the model (seconds)
Naïve Bayes Classifier	BOW	78.80%	0.25	0.189
	TFIDF	75.10%	0.26	0.193
Support Vector Mechanism	BOW	90.15%	0.90	14.840
	TFIDF	90.20%	0.90	11.280
Generalized Logistic Regression	BOW	89.35%	0.90	22.400
	TFIDF	89.65%	0.90	7.362
Decision Tree	BOW	86.40%	0.87	1.877
	TFIDF	86.70%	0.87	1.788
Random Forest	BOW	88.35%	0.88	19.705
	TFIDF	89.10%	0.89	16.575

After hyperparameter tuning, the SVM using BOW and TFIDF features remained the top performers, achieving high accuracy and weighted F1-scores. The Generalized Logistic Regression models, both before and after hyperparameter tuning, also achieved competitive performance. Overall, based on the provided results, the SVM with either BOW or TFIDF features appeared to be the best-performing model, with high accuracy and weighted F1-score. The Confusion matrix and the SVM model results after hyperparameter tuning are shown in TABLE IV, Fig. 5, TABLE V and Fig. 6.

TABLE IV. Classification Report of SVM with BOW After Hyperparameter Tuning

	Precision	Recall	F1-score	Support
Anger	0.89	0.90	0.90	275
Fear	0.87	0.88	0.88	224
Joy	0.93	0.92	0.92	695
Love	0.75	0.81	0.78	159
Sadness	0.95	0.94	0.94	581
Surprise	0.74	0.70	0.72	66
Accuracy			0.90	2000
Macro Avg	0.86	0.86	0.86	2000
Weighted Avg	0.90	0.90	0.90	2000

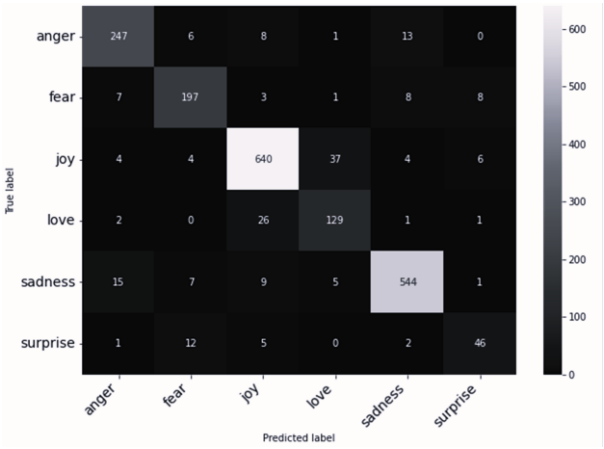

Fig. 5. Results of SVM with BOW after hyperparameter tuning

TABLE V. Classification Report of SVM with TF-IDF After Hyperparameter Tuning

	Precision	Recall	F1-score	Support
Anger	0.89	0.90	0.90	275
Fear	0.90	0.80	0.85	224
Joy	0.90	0.96	0.93	695
Love	0.89	0.69	0.78	159
Sadness	0.94	0.95	0.94	581
Surprise	0.67	0.67	0.67	66
Accuracy			0.90	2000
Macro Avg	0.86	0.83	0.84	2000
Weighted Avg	0.90	0.90	0.90	2000

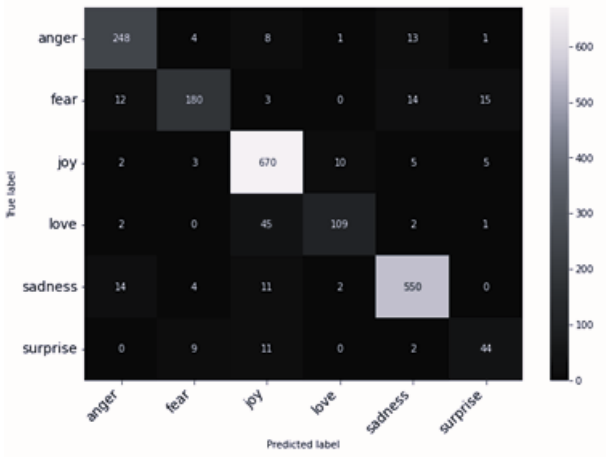

Fig. 6. Results of SVM with TFIDF after hyperparameter tuning

The SVM models using BOW and TF-IDF features achieved similar high accuracy of around 0.90. The models performed well across most classes, with high precision, recall, and F1-scores. The "joy" and "sadness" classes consistently performed well in both models. The TF-IDF based SVM model had slightly lower precision, recall, and F1-scores for the "fear" and "love" classes than the BOW-based model. Overall, both models demonstrated strong performance in classifying emotions, with the SVM model using BOW features having a slight edge in precision and recall.

The deep learning model was created with LSTM-based RNN with embedding, dropout, LSTM and dense layers. The model is trained for a specific number of epochs using early stopping, based on the validation loss. Training a model can be computationally expensive and due to the limited resources, empirically it was derived that 5 is the best starting point for epoch in this scenario. But, it is also acknowledged that this might not be the optimal number and further exploration is needed. The loss and accuracy values are tested, and training and validation loss/accuracy curves are plotted as in Fig. 7.

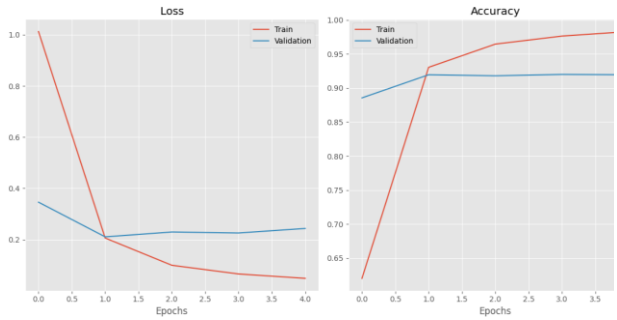

Fig. 7. Loss and accuracy function plots for training and validation sets

The loss plot shows the change in loss values over the epochs for both the training and validation sets. The training loss shows how well the model is fitting the training data while validation loss shows how well the model is fitting the validation data, which is untouched. Since the training loss is lower than validation loss, it may suggest that the model is overfitting. This is applicable for accuracy plot as well. The accuracy plot shows the change in accuracy values over the epochs for both the training and validation sets. The training accuracy shows how well the model is fitting the training data while validation accuracy shows how well the model is fitting the validation data. The training accuracy seems higher than the validation accuracy, which maybe an indication that the model is overfitting. Nevertheless, the confusion matrix is generated for the above output, as shown in Fig. 8.

TABLE VI. CLASSIFICATION REPORT OF THE MODEL AFTER HYPERPARAMETER TUNING

	Precision	Recall	F1-score	Support
Anger	0.91	0.93	0.92	275
Fear	0.93	0.83	0.88	224
Joy	0.94	0.94	0.94	695
Love	0.80	0.84	0.82	159
Sadness	0.96	0.96	0.96	581
Surprise	0.76	0.88	0.82	66
Accuracy			0.92	2000
Macro Avg	0.88	0.90	0.89	2000
Weighted Avg	0.92	0.92	0.92	2000

Fig. 8. Confusion matrix of the model after hyperparameter tuning

The model's overall accuracy resulted in an accuracy level of 92.2%. The macro-average F1-score, precision, and recall across all classes were 0.89, 0.88, and 0.90, respectively. The weighted average F1-score, precision, and recall were all 0.92, indicating good overall performance for the model.

V. DISCUSSION

The machine learning models experimented with are Naïve Bayes, Support Vector Mechanism, Regression, Decision Tree, and Random Forest. High accuracy was achieved after following basic steps such as BOW and TF-IDF. It shows that to accomplish higher accuracy, it is not always required to use complex algorithms. After tuning the hyperparameters, the models were evaluated again using the validation set. Once the hyperparameters were tuned, the models were evaluated on the hold-out set of data.

Since the dataset is biased (i.e., The number of records that fall into each label is different), the weighted average for F1-score was used to assess the model performances. Another metric that is used to measure the model performance is the time taken to train the model. It is noticed that models (except

for the SVM model) took more time to train when the BOW was used as the feature extraction method instead of the TF-IDF method. The Random Forest model took the highest time to train. It achieved an accuracy level of 89%, while the Naïve Bayes model took less than one second to train, although its accuracy was comparatively low. When the cost factor plays a big role in modeling, one can select an average-performing model which takes less time to train. As the objective of a machine learning model is to maximize accuracy and minimize the computation time required for training, the models above are preferred.

The deep learning model was fitted on the data using Keras, a Python-based neural network library. The model type that is used is sequential modeling which has allowed us to build a linear stack of layers. LSTM network is implemented on the data, which is an RNN [21]. When the model's performance is evaluated, it is visible that the deep learning model outperforms all the machine learning models. The weighted average for F1-score was 92%, while the model's accuracy was also 92%.

One of the limitations one faces while working on emotion analysis in the text is less availability of emotion-annotated data. Furthermore, the available data are in the English language. For hyperparameter tuning in the machine learning models, the GridSearch method is proposed due to the literature review. Some approaches, such as RandomSearch, and Bayesian Optimization, would have been applied to the models to identify the method that performed well on the chosen dataset. As stated by [18] another problem a researcher faces when detecting emotions from text is the absence of facial expressions and voice modulations.

Furthermore, the lack of contextual understanding could lead to a change in emotion classification. One of the best examples to prove this statement is the study carried out by [22]. They proposed a system to aid children to detect insulting words and it resulted that ignoring contextual information resulted in misclassification.

The growing use of Internet slang has hindered sentiment analysis research. For example, slang such as 'lol' stands for laughing out loud, and 'bussin" is the new way of explaining that something is good. Furthermore, there is a tendency to explain anger in the form of sarcasm. Using machine learning algorithms, this is not easy to detect, although humans can judge it easily. Another challenge that researchers in this domain face is that one sentence could contain multiple emotions. For example, let us take the sentence, "Customer service at restaurant one was worse than the service at restaurant two." To the human brain, this gives the idea that restaurant two's service was better than the service provided at restaurant one. Nevertheless, a machine understands this negatively because of the word "worse." Although it is used positively, since the word "worse" is used negatively, the whole sentence is categorized negatively.

Additional research endeavors could involve quantifying the intensity of emotions alongside the examination of various emotion types, thereby enhancing the analysis of emotion classification. Since discrete emotional models have been used for this experiment, neither the emotion's intensity nor the emotion's valance is considered. Humans feel a spectrum of emotions; thus, this research analyzes to which percentage one aspect of emotion is aroused. For example, one can feel angry and sad simultaneously in a given situation. Therefore, identifying to which degree one emotion has aroused is another aspect that can be considered.

REFERENCES

[1] L. Canales and P. Martínez-Barco, “Emotion Detection from text: A Survey,” in *Proceedings of the Workshop on Natural Language Processing in the 5th Information Systems Research Working Days (JISIC)*, Quito, Ecuador: Association for Computational Linguistics, 2014, pp. 37–43. doi: 10.3115/v1/W14-6905.

[2] A. Hassan Yousef, W. Medhat, and H. Mohamed, “Sentiment Analysis Algorithms and Applications: A Survey,” *Ain Shams Engineering Journal*, vol. 5, May 2014, doi: 10.1016/j.asej.2014.04.011.

[3] Y. Chew-Yean, “Emotion Detection and Recognition from Text Using Deep Learning,” CSE Developer Blog. Accessed: Mar. 09, 2022. [Online]. Available: https://devblogs.microsoft.com/cse/2015/11/29/emotion-detection-and-recognition-from-text-using-deep-learning/

[4] F. A. Acheampong, C. Wenyu, and H. Nunoo-Mensah, “Text-based emotion detection: Advances, challenges, and opportunities,” *Engineering Reports*, vol. 2, no. 7, p. e12189, 2020, doi: 10.1002/eng2.12189.

[5] K. Jones, K. Libert, and K. Tynski, “The Emotional Combinations That Make Stories Go Viral,” *Harvard Business Review*, May 23, 2016. Accessed: Apr. 10, 2022. [Online]. Available: https://hbr.org/2016/05/research-the-link-between-feeling-in-control-and-viral-content

[6] P. Nandwani and R. Verma, “A review on sentiment analysis and emotion detection from text,” *Soc. Netw. Anal. Min.*, vol. 11, no. 1, p. 81, Aug. 2021, doi: 10.1007/s13278-021-00776-6.

[7] S. Kusal, S. Patil, K. Kotecha, R. Aluvalu, and V. Varadarajan, “AI Based Emotion Detection for Textual Big Data: Techniques and Contribution,” *BDCC*, vol. 5, no. 3, p. 43, Sep. 2021, doi: 10.3390/bdcc5030043.

[8] M. Gamon, “Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis,” in *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, Geneva, Switzerland: COLING, Aug. 2004, pp. 841–847. Accessed: Nov. 20, 2022. [Online]. Available: https://aclanthology.org/C04-1121

[9] Q. Ye, Z. Zhang, and R. Law, “Sentiment classification of online reviews to travel destinations by supervised machine learning approaches,” *Expert Systems with Applications*, vol. 36, no. 3, Part 2, pp. 6527–6535, Apr. 2009, doi: 10.1016/j.eswa.2008.07.035.

[10] B. Desmet and V. Hoste, “Emotion detection in suicide notes,” *Expert Systems with Applications*, vol. 40, no. 16, pp. 6351–6358, Nov. 2013, doi: 10.1016/j.eswa.2013.05.050.

[11] L. D. C. S. Subhashini, Y. Li, J. Zhang, A. S. Atukorale, and Y. Wu, “Mining and classifying customer reviews: a survey,” *Artif Intell Rev*, vol. 54, no. 8, pp. 6343–6389, Dec. 2021, doi: 10.1007/s10462-021-09955-5.

[12] H. Pouransari and S. Ghili, “Deep learning for sentiment analysis of movie reviews,” p. 8.

[13] Y. Wang, S. Feng, D. Wang, G. Yu, and Y. Zhang, “Multi-label Chinese Microblog Emotion Classification via Convolutional Neural Network,” in *Web Technologies and Applications*, vol. 9931, F. Li, K. Shim, K. Zheng, and G. Liu, Eds., in Lecture Notes in Computer Science, vol. 9931. , Cham: Springer International Publishing, 2016, pp. 567–580. doi: 10.1007/978-3-319-45814-4_46.

[14] L. Zhang, S. Wang, and B. Liu, “Deep learning for sentiment analysis: A survey,” *WIREs Data Mining Knowl Discov*, vol. 8, no. 4, Jul. 2018, doi: 10.1002/widm.1253.

[15] M. Jabreel and A. Moreno, “A Deep Learning-Based Approach for Multi-Label Emotion Classification in Tweets,” *Applied Sciences*, vol. 9, no. 6, Art. no. 6, Jan. 2019, doi: 10.3390/app9061123.

[16] S. Mohammad and F. Bravo-Marquez, “Emotion Intensities in Tweets,” in *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 65–77. doi: 10.18653/v1/S17-1007.

[17] “EmoInt-web2,” Tableau Software. Accessed: Mar. 09, 2022. [Online]. Available: https://public.tableau.com/views/EmoInt-web2/EmotionIntensityDashboard?:embed=y&:showVizHome=no&:host_url=https%3A%2F%2Fpublic.tableau.com%2F&:tabs=no&:toolbar=yes&:animate_transition=yes&:display_static_image=no&:display_spinner=no&:display_overlay=yes&:display_count=yes&publish=yes&:loadOrderID=0

[18] A. Chatterjee, K. N. Narahari, M. Joshi, and P. Agrawal, "SemEval-2019 Task 3: EmoContext Contextual Emotion Detection in Text," in *Proceedings of the 13th International Workshop on Semantic Evaluation*, Minneapolis, Minnesota, USA: Association for Computational Linguistics, Jun. 2019, pp. 39–48. doi: 10.18653/v1/S19-2005.

[19] L. D. C. S. Subhashini, Y. Li, J. Zhang, and A. S. Atukorale, "Integration of Fuzzy and Deep Learning in Three-Way Decisions," in *2020 International Conference on Data Mining Workshops (ICDMW)*, Sorrento, Italy: IEEE, Nov. 2020, pp. 71–78. doi: 10.1109/ICDMW51313.2020.00019.

[20] E. Saravia, H.-C. T. Liu, Y.-H. Huang, J. Wu, and Y.-S. Chen, "CARER: Contextualized Affect Representations for Emotion Recognition," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 3687–3697. doi: 10.18653/v1/D18-1404.

[21] "Understanding LSTM Networks -- colah's blog." Accessed: Feb. 25, 2023. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[22] M. Allouche, A. Azaria, R. Azoulay, E. Ben-Izchak, M. Zwilling, and D. Zachor, "Automatic Detecting of Insulting Sentences in Conversation," Jan. 2019. doi: 10.1109/ICSEE.2018.8646165.